PATENT

CERTIFICATE OF MAILING UNDER 37 C.F.R. § 1.8
I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on August 3, 2005.

David G. Parkhurst

David G. Parkhurst, Reg. No. 29,422

OIPE AUG 08 2005

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

MASAAKI OKA, ET AL.

Serial No. 09/630,248

Filed: August 1, 2000

For: PICTURE DRAWING APPARATUS AND PICTURE DRAWING METHOD

Examiner: Scott A. Wallace

Group Art Unit 2675

Docket No. SUZCO 55325

August 3, 2005

Los Angeles, California 90045

SUBMISSION OF CERTIFIED TRANSLATION OF PRIORITY DOCUMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Enclosed herewith is a certified translation of the Japanese Patent Application No. 8- 020332 which was submitted as a priority document in connection with PCT/JP97/00297.

By David G. Parkhurst
David G. Parkhurst
Registration No. 29,422

DGP:rvw
Encls. Return Postcard

Howard Hughes Center
6060 Center Drive, Tenth Floor
Los Angeles, CA 90045
Telephone: (310) 824-5555
Facsimile: (310) 824-9696

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT :
APPLICATION No. :
FILING DATE :
TITLE :

Group Art Unit :
Examiner :

Hon. Commissioner of Patents and Trademarks,
Washington, D.C. 20231

SIR:

CERTIFIED TRANSLATION

I, Kouchi Takasaki, am an official translator of the Japanese language into the English language and I hereby certify that the attached comprises an accurate translation into English of Japanese Application No.8-020332, filed on February 6, 1996.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

10/9/1997
Date

Kouchi Takasaki
Kouchi Takasaki

Patent Office

Japanese Government

This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: February 6, 1996

Application Number: Patent Application
Ser. No.8-020332

Applicant: Sony Computer Entertainment Inc.

March 28, 1997

Commissioner,
Patent Office Hisamitsu Arai

[Document Name]	Patent Application
[Reference Number]	9600790202
[Filing Date]	February 6, 1996
[To]	Hon.Commissioner,Patent Office
[IPC]	G06T 11/40
[Title of the Invention]	Picture Drawing Apparatus and Picture Drawing Method
[Number of Claims]	4
[Inventor]	
[Address]	c/o Sony Computer Entertainment Inc., 11-22, Akasaka 8-chome, Minato-ku, Tokyo, Japan
[Name]	Toshiyuki Hiroi
[Address]	c/o Sony Computer Entertainment Inc., 11-22, Akasaka 8-chome, Minato-ku, Tokyo, Japan
[Name]	Masaaki Oka
[Patent Applicant]	
[Identification Number]	395015319
[Name]	Sony Computer Entertainment Inc.
[Representative]	Teruhisa Tokunaka
[Patent Attorney]	
[Identification Number]	100067736
[Patent Attorney]	
[Name]	Akira Koike
[Patent Attorney]	
[Identification Number]	100086335
[Patent Attorney]	
[Name]	Eiichi Tamura
[Patent Attorney]	
[Identification Number]	100096677
[Patent Attorney]	
[Name]	Seiji Iga
[Indication of Charge]	
[Method of Payment]	Deposit
[Number of Prepaid Ledger]	019530

[Amount]	21,000 yen	
[List of Document]		
[Document]	Specification	1
[Document]	Drawing	1
[Document]	Summary	1
[General Power of Attorney Number]		9506905
[Need of Proof]	Yes	

[Name of Document] SPECIFICATION

[Title of the Invention]

Picture Drawing Apparatus and Picture Drawing Method

[Claims]

[Claim 1]

A picture drawing apparatus comprising:

pre-processing means for performing pre-processing to generate data required for drawing processing on the unit figure basis in accordance with a drawing command for drawing a picture model defined by a combination of unit figures;

picture drawing means for generating pixel data on the unit figure basis by texture mapping based on the data provided from the pre-processing means for drawing a picture on a picture memory; and

texture cache for transiently storing texture data required by the picture drawing means for texture mapping;

the texture data required by the picture drawing means for texture mapping being transferred to the texture cache on the pre-processing by the pre-processing means, the pre-processing means and the picture drawing means being operated by pipelining.

[Claim 2]

The picture drawing apparatus as claimed in claim 1, wherein the picture drawing means has the function of performing MIP mapping, and

data having a resolution required for the picture drawing

means to perform MIP mapping is selected from texture data on a texture memory on the pre-processing by the pre-processing means and transferred to the texture cache.

[Claim 3]

A picture drawing method comprising the steps of performing pre-processing to generate data required for picture drawing processing on the unit figure basis in accordance with a drawing command for drawing a picture model defined by the combination of unit figures, generating pixel data of a unit figure by texture mapping on the basis of the data obtained by the pre-processing, and performing picture drawing processing to draw a picture on a picture memory,

texture data required for texture mapping in the picture drawing processing being transferred to the texture cache from a texture memory on the pre-processing, the pre-processing and the picture drawing processing being performed by pipelining.

[Claim 4]

The picture drawing method as claimed in claim 3, wherein in generating pixel data of a unit figure by MIP mapping in the picture drawing processing for performing picture drawing processing to draw a picture on the picture memory,

data having a resolution required for MIP mapping in the picture drawing processing is selected from texture data on the texture memory on the pre-processing and transferred to the texture cache.

[Detailed Description of the Invention]

[0001]

[Technical Field to which the Invention Pertains]

This invention relates to a picture drawing apparatus and a picture drawing method used in a graphics computer, a special effect device or a video game machine which are a video equipment employing computers.

[0002]

[Prior Art]

Conventionally, in a picture generating device in a domestic TV game machine, personal computers or graphics computers, used for generating data of a picture outputted for display to a television receiver, a monitor receiver or a cathode ray tube (CRT) display device, a dedicated drawing device is provided between central processing unit (CPU) and a frame buffer for enabling high-speed processing.

[0003]

That is, in the above-mentioned picture generating device, the CPU when generating a picture does not directly access the frame buffer, but performs geometry processing, such as coordinate transformation, clipping or light source calculations, defines a three-dimensional model as combination of basic unit figures, such as triangles or quadrangles, in order to formulate a drawing command for drawing a three-dimensional picture, and sends the drawing command thus generated to the drawing device.

For generating, for example, a three-dimensional object, the object is resolved into plural polygons and a drawing command for each polygon is sent from the CPU to the drawing device. The drawing device interprets the drawing command sent from the CPU to execute rendering processing of writing pixel data in the frame buffer, taking into account the Z-values and colors of all pixels making up the polygon, from the Z-values specifying color data and depth of the apex points, for drawing a figure on the frame buffer. The Z-values represent the information specifying the distance from the viewing point along the depth direction.

[0004]

If, for example, a three-dimensional object is displayed in the above-described picture generating device, the object is resolved into plural polygons, and a drawing command associated with each polygon is transmitted from the CPU to the drawing device. For representing the object more realistically, there are used techniques termed texture mapping or MIP mapping. There are also widely known techniques of converting picture color data via a color lookup table (CLUT) having stored therein color conversion data for changing the display colors.

[0005]

The texture mapping is a technique of affixing a two-dimensional picture pattern provided separately as a texture source picture to the surface of a polygon constituting the object. The MIP mapping, on the other hand, is among the

techniques of texture mapping of interpolating pixel data so that the affixing pattern on the polygon will not be non-spontaneous in case the three-dimensional model is moved towards or away from the viewing point.

[0006]

[Problem to be Solved by the Invention]

Meanwhile, the picture drawing speed depends on the processing speed of texture mapping or MIP mapping for each polygon in a drawing engine. Moreover, the picture drawing speed is influenced by the writing speed from the drawing engine to the frame buffer, such that, if the frame buffer accessing speed is low, the drawing speed is lowered. Therefore, if an expensive high-speed memory is used as a large-capacity frame buffer, for increasing the drawing speed, the system cost is prohibitively increased. However, if an inexpensive dynamic random-access memory (DRAM) is used, the drawing speed of the system is lowered.

[0007]

In view of the above depicted status of the art, the present invention has for its objects to provide the following.

[0008]

That is, it is an object of the present invention to provide a picture drawing device and a picture drawing method whereby a high drawing speed may be maintained even with the use of an inexpensive memory, such as DRAM, as a frame buffer.

[0009]

It is another object of the present invention to provide a picture drawing device and a picture drawing method whereby, in the picture drawing device for executing texture mapping by drawing means, drawing processing can be executed without stopping the drawing means.

[0010]

It is yet another object of the present invention to provide a picture drawing device and a picture drawing method whereby the number of times of accessing and the accessing time of the picture memory can be decreased to raise the overall picture drawing speed.

[0011]

[Means to Solve the Problem]

According to the present invention, there is provided a picture drawing apparatus comprising: pre-processing means for performing pre-processing to generate data required for drawing processing on the unit figure basis in accordance with a drawing command for drawing a picture model defined by a combination of unit figures; picture drawing means for generating pixel data on the unit figure basis by texture mapping based on the data provided from the pre-processing means for drawing a picture on a picture memory; and texture cache for transiently storing texture data required by the picture drawing means for texture mapping; the texture data required by the picture drawing means

for texture mapping being transferred to the texture cache on the pre-processing by the pre-processing means, the pre-processing means and the picture drawing means being operated by pipelining.

[0012]

In the picture drawing apparatus according to the present invention, the picture drawing means has the function of performing MIP mapping, and data having a resolution required for the picture drawing means to perform MIP mapping is selected from texture data on a texture memory on the pre-processing by the pre-processing means and transferred to the texture cache.

[0013]

According to the present invention, there is also provided a picture drawing method comprising the steps of performing pre-processing to generate data required for picture drawing processing on the unit figure basis in accordance with a drawing command for drawing a picture model defined by the combination of unit figures, generating pixel data of a unit figure by texture mapping on the basis of the data obtained by the pre-processing, and performing picture drawing processing to draw a picture on a picture memory, in which texture data required for texture mapping in the picture drawing processing is transferred to the texture cache from a texture memory on the pre-processing, and the pre-processing and the picture drawing processing is performed by pipelining.

[0014]

In the picture drawing method according to the present invention, in generating pixel data of a unit figure by MIP mapping in the picture drawing processing for performing picture drawing processing to draw a picture on the picture memory, data having a resolution required for MIP mapping in the picture drawing processing is selected from texture data on the texture memory on the pre-processing and transferred to the texture cache.

[0015]

[Mode for Carrying Out the Invention]

Referring to the drawings, preferred embodiments of the present invention will be explained in detail.

[0016]

The drawing device according to the present invention is applied to a video game device as shown in Fig.1. The drawing method according to the present invention is carried out on this video game device.

[0017]

The video game machine executes a game in accordance with instructions by the user by reading and executing a game program stored in an auxiliary memory device such as an optical disc, and has a configuration as shown in Fig.1.

[0018]

Specifically, the present video game device has two sorts of buses, namely a main bus 1 and a sub-bus 2.

[0019]

The main bus 1 and the sub-bus 2 are interconnected via a bus controller 10.

[0020]

To the main bus 1 are connected a central processing unit (CPU) 11, made up of a micro-processor, a main memory 12, made up of a random access memory (RAM), a main dynamic memory access controller or main DMAC 13, a MPEG decoder 14 and a picture processing unit or graphic processing unit (GPU) 15. To the sub-bus 2 are connected a subsidiary central processing unit or sub-CPU 21, made up of a micro-processor, a subsidiary memory, made up of a random access memory (RAM), a subsidiary dynamic memory accessing controller or sub DMAC 23, a read-only memory (ROM) 24, having stored therein a program, such as an operating system, a sound processing unit (SPU) 25, a communication controller or asynchronous transmission mode (ATM) 26, an auxiliary storage device 27, and an input device 28.

[0021]

The bus controller 10 is a device on the main bus 1 for switching between the main bus 1 and the sub-bus 2, and is opened in an initial state.

[0022]

The main CPU 11 is a device on the main bus 1 operating by a program on the main memory 12. Since the bus controller 10 is open on start-up, the main CPU 11 reads in a boot program from

the ROM 24 on the sub-bus 2 to reproduce the application program and necessary data from the auxiliary storage device 27 for loading on the main memory 12 or on devices on the sub-bus 2. On the main CPU 11 is loaded a geometry transfer engine (GTE) 17 for performing processing such as coordinate transformation. This GTE 17 has a parallel computing mechanism for carrying out plural computing processing operations in parallel and is responsive to a request for computing processing from the CPU 11 in order to carry out fast processing operations, such as coordinate transformation, light source calculations, matrix or vector operations. Based on the results of computing processing operations by the GTE 17, the main CPU 11 defines a three-dimensional model as a combination of basic unit figures, such as triangles or quadrangles, formulates a drawing command associated with each polygon for drawing a three-dimensional picture and packetizes the drawing command in order to route the resulting command packet to the GPU 15.

[0023]

The main DMAC 13 is a device on the main bus 1 for performing control such as DMA transfer on the devices on the main bus 1. If the bus controller 10 is open, the main DMAC 13 also controls the devices on the sub-bus 2.

[0024]

The CPU 15 is a device on the main bus 1 functioning as a rendering processor. This GPU 15 interprets the drawing command

sent from the main CPU 11 or main DMAC 13 as command packet and performs rendering processing of writing pixel data in the frame buffer 18 from Z-values specifying depth and color data of all pixels making up a polygon.

[0025]

The MDEC 14 is an I/O connection device operable in parallel with the main CPU 11 and is a device functioning as a picture expansion engine. The MPEG decoder (MDEC) 14 decodes picture data compressed and encoded by orthogonal transform, such as discrete cosine transform.

[0026]

On the sub-bus 2, the sub-CPU 21 is a device on the sub-bus 2 operating in accordance with a program on the sub-memory 22.

[0027]

The sub-DMAC 23 is a device on the sub-bus 2 for performing control such as DMAC transfer for devices on the sub-bus 2. This sub-DMAC 23 can acquire bus rights only when the bus controller 16 is closed.

[0028]

The SPU 25 is a device on the sub-bus 2 functioning as a sound processor. This SPU 25 is responsive to a sound command sent as a command packet from the sub-CPU 21 or sub-DMAC 23 as a command packet to read out sound source data from the sound memory 29 to output the read-out data.

[0029]

The ATM 26 s a communication device on the sub-bus 2.

[0030]

The auxiliary storage device 27 is a data input device on the sub-bus 2 and is made up of a disk drive or the like.

[0031]

The input device 28 is an input device from other equipments, such as a control pad on the sub-bus 2, man-machine interface, picture input or speech input.

[0032]

That is, in the above-described video game machine, the geometry processing system, executing geometry processing, such as coordinate transformation, clipping or light source calculations, formulating a drawing command for defining a three-dimensional model as combination of basic unit figures (polygons) such as triangles or quadrangles for drawing a three-dimensional picture, and sending out the drawing command associated with each polygon as command packet to the main bus 1, is made up of the main CPU 11 and the GTE 17 on the main bus 1. Also, in the video game machine, the rendering processing system for formulating pixel data of each polygon based on the drawing command from the geometry processing system for writing a figure in the frame buffer 18 by way of rendering processing for drawing a figure in the frame buffer 18 is constituted by the GPU 15.

[0033]

The above-mentioned GPU 15 is explained in detail.

[0034]

Referring to Fig.2, the GPU 15 includes a packet engine 31 connected to the main bus 1 shown in Fig.1 and performs rendering processing of writing pixel data of each polygon in the frame buffer 18 by the pre-processor 32 and the drawing engine 33 in accordance with the drawing command sent as a command packet from the main CPU 11 or main DMAC shown in Fig.1 over the main bus 1 to the packet engine 31, reading out pixel data of the picture drawn in the frame buffer 18 and furnishing the read-out pixel data via CRT controller 34 as video signals to a television receiver or a monitor receiver, not shown.

[0035]

The packet engine 31 develops the command packet sent from the main CPU 11 or main DMAC 13 shown in Fig.1 over the main bus 1 on a register, not shown.

[0036]

The pre-processor 32 generates polygon data in accordance with the drawing command sent to the packet engine 31 as command packet and performs pre-set pre-processing, such as polygon division as later explained, on the polygon data, while generating various data such as apex point coordinate information for the respective polygons required by the drawing engine 33, address information such as texture or MIP map texture, or control information, such as pixel interleaving.

[0037]

The drawing engine 33 includes N polygon engines 33A1, 33A2, ..., 33AN, connected to the pre-processor 32, N texture engines 33B1, 33B2, ..., 33BN, connected to the polygon engines 33A1, 33A2, ..., 33AN, a first bus switcher 33C, connected to the texture engines 33B1, 33B2, ..., 33BN, M pixel engines 33D1, 33D2, ..., 33DM, connected to the first bus switcher 33C, a second bus switcher 33E, connected to the pixel engines 33D1, 33D2, ..., 33DM, a texture cache 33F connected to the second bus switcher 33E, and a CLUT cache 33G connected to the texture cache 33F.

[0038]

In the drawing engine 33, the N polygon engines 33A1, 33A2, ..., 33AN sequentially generate polygons in accordance with the drawing command on the basis of the polygon data pre-processed by the pre-processor 32 for performing parallel shading processing from one polygon to another.

[0039]

The N texture engines 33B1, 33B2,..., 33BN perform texture mapping or MIP Mapping in parallel, based on the texture data supplied from the texture cache 33F via color lookup table (CLUT) cache 33G, on each of the polygons generated by the polygon engines 33A1, 33A2, ..., 33AN.

[0040]

The pre-processor 32 previously furnishes the address information, such as texture or MIP MAP texture, affixed to the

polygons processed by the N texture engines 33B1, 33B2, ..., 33BN, to the texture cache 33F. Based on the above address information, the texture data necessary for texture mapping is transferred from the texture area on the frame buffer 18, while only resolution data necessary for MIP Mapping is selected from the relevant texture data so as to be transferred as MIP MAP texture data. To the CLUT cache 33G CLUT data to be referred to during texture drawing is transferred from the CLUT area on the frame buffer 18.

[0041]

The polygon data, processed with texture mapping or MIP Mapping by the N texture engines 33B1, 33B2, ..., 33BN, are transferred via first bus switcher 33C to M pixel engines 33D1, 33D2, ..., 33DM.

[0042]

The M pixel engines 33D1, 33D2, ..., 33DM perform various picture processing operations, such as Z-buffer processing or anti-aliasing processing, in parallel, for generating M pixel data.

[0043]

The M pixel data, generated by the M pixel engines 33D1, 33D2, ..., 33DM, are written in the frame buffer 18 via second bus switcher 33E.

[0044]

The second bus switcher 33E is fed with the pixel

interleaving control information from the pre-processor 32. The second bus switcher 33E has the function of performing pixel interleaving processing of selecting L pixel data from the M pixel data generated by the M pixel engines 33D1, 33D2, ..., 33DM in accordance with the above control information for writing M pixel data with M storage locations conforming to the shape of the polygon drawn on the frame buffer 18 as accessing unit.

[0045]

The drawing engine 33 generates and writes all pixel data of each polygon in the frame buffer 18, based on the polygon data pre-processed by the pre-processor 32, for drawing a picture defined as the combination of the polygons by the drawing command on the frame buffer 18. Thus the pixel data of the picture drawn on the frame buffer 18 is read out by the second bus switcher 33E so as to be furnished via CRTC 34 as video signals to a television receiver or to a monitor receiver, not sown.

[0046]

In the above-described structure of the GLUT 15, the pre-processor 32 generates the address information for pre-reading the texture affixed to the polygon processed by the N texture engines 33B1, 33B2, ..., 33BN, ..., based on the apex point coordinates [(X0, Y0), (X1, Y1), (X2, Y2)] of the polygons and the texture coordinates [(U0, V0), (U1, V1), (U2, V2)]. In addition, the GLUT 15 reproduces the MIP mapping selection information from the tilt of the sides of the polygons [(X1-

X0)/(Y1−Y0), [(X2−X0)/(Y2−Y0), [(X1−X2)/(Y1−Y2)], tilt of the texture addresses [(U1−U0)/(Y1−Y0), [(U2−U0)/(Y2−Y0), [(U1−U2)/(Y1−Y2), [(V1−V0)/(Y1−Y0), [(V2−V0)/(Y2−Y0), [(V1−V2)/(Y1−Y2), ..., or the polygon areas, to furnish the information to the texture cache 33F. The GLUT 15 sorts the polygon apex point coordinates [(X0, Y0), (X1, Y1), (X2, Y2)] in the sequence of the apex points of the left edges (X0, Y0) → (X1, Y1) → (X2, Y2) or in the sequence of the apex points of the right edges (X2, Y2) → (X1, Y1) → (X0, Y0) or scans both end points or texture addresses.

[0047]

The pre-processor 32 stores the information corresponding to the pre-processed polygon data in a work memory, not shown. At a stage in which the drawing engine 33 can process the next polygon, the information capable of processing one polygon is transferred from the work memory to the N polygon engines 33A1, 33A2, ..., 33AN. This causes the drawing engine 33 to start drawing processing for a new polygon.

[0048]

That is, with the present GPU 15, the pre-processor 32 and the drawing engine 33 executes drawing processing by pipelining processing for drawing a picture defined as the combination of the polygons under a drawing command.

[0049]

The drawing processing by this pipelining processing is

again explained.

[0050]

The pre-processor 32 performs pre-set pre-processing on the polygon data as described above and furnishes various data, such as the apex point coordinate information required by the drawing engine 33, address information for texture or MIP mapping texture or control information, such as pixel interleaving, to the drawing engine 33.

[0051]

The drawing engine 33 receives data from the pre-processor 32 to read out the necessary texture data from the texture cache 33D to generate data in order to write the generated pixel data in the frame buffer 18. The texture cache 33D reads out texture data of a texture area corresponding to the necessary texture addresses calculated by pre-processing in the pre-processor 32 from the frame buffer 18. The texture data is read out so that data readout will be completed before start of picture drawing employing the texture data. The number of times of accessing to the texture area can be decreased by reading only the texture data corresponding to the resolution required for MIP mapping from the texture area.

[0052]

The data structure in the texture cache 33F, shown as an example in Fig.4, is comprised of a tag area TAG made up of texture addresses, a storage area DATA having stored therein the

necessary texture data and a flag L specifying that the texture data has not as yet been used. For employing the entry, having the flag L reset, the texture cache 33 reads in the texture data from the texture area of the frame buffer 18 to set its flag L. The drawing engine 33 reads out the corresponding texture data from the entry, the flag of which has been set, in order to perform drawing processing, and resets the flag 1 of the entry at a stage in which the drawing has come to a close and hence the texture data is no longer required.

[0053]

In the drawing device, in which the texture mapping processing is carried out as described above, the pre-processor 32 and the drawing engine 33 are constructed as pipeline such that the texture data required by the drawing engine 33 are sent from the texture memory, that is the texture area on the frame buffer 18, to the cache memory 33F at a pre-proceeding stage by the pre-processor 32, so that the drawing processing can be carried out without stopping the drawing engine 33. In addition, by reading only the texture data associated with the resolution required for MIP mapping from the texture area, the number of times of accessing to the texture area can be diminished to raise the drawing speed of the entire device.

[0054]

The polygon division processing by the pre-processor 32 is carried out in accordance with a flowchart shown in Fig.5.

[0055]

Specifically, the polygon division processing is started with the polygon count C specifying the number of polygons set to 1.

[0056]

At a first processing step S1, it is judged whether or not it is necessary to divide a polygon. In the judgment processing in the processing step S1, it is judged whether or not the polygon now processed is comprised within the range of the texture cache 33F. For this judgment processing, it suffices if calculated values of texture coordinates [(U0, V0), (U1, V1), (U2, V2)] of the apex points of the polygon are within one texture page.

[0057]

If the result of judgment at the processing step S1 is NO, that is if the polygon needs to be divided, processing transfers to step S2 to perform polygon division by N. Such division by N of the polygon at this processing step S2 is carried out by dividing all sides of the polygon at neutral points, as shown below:

[0058]

$$X0' = (X0 + X1)/2$$

$$Y0' = (Y0 + Y1)/2$$

$$Z0' = (Z0 + Z1)/2$$

$$X1' = (X1 + X2)/2$$

$Y1' = (Y1 + Y2)/2$

$Z1' = (Z1 + Z2)/2$

$X2' = (X2 + X0)/2$

$Y2' = (Y2 + Y0)/2$

$Z2' = (Z2 + Z0)/2$

$U0' = (U0 + U1)/2$

$V0' = (V0 + V1)/2$

$Z0' = (Z0 + Z1)/2$

$U1' = (U1 + U2)/2$

$V1' = (V1 + V2)/2$

$Z1' = (Z1 + Z2)/2$

$U2' = (U2 + U0)/2$

$V2' = (V2 + VO)/2$

$Z2' = (Z2 + Z0)/2$

$R0' = (R0 + R1)/2$

$G0' = (G0 + G1)/2$

$B0' = (B0 + B1)/2$

$R1' = (R1 + R2)/2$

$G1' = (G1 + G2)/2$

$B1' = (B1 + B2)/2$

$R2' = (R2 + R0)/2$

$G2' = (G2 + G0)/2$

$B2' = (B2 + B0)/2$

That is, in the division by N of the polygon in the processing step S2, all sides of the polygon are divided at

middle points thereof for dividing a triangular polygon, for example, into new polygons with N = 4.

[0059]

At the next processing step S2, the number of polygons is changed by setting the polygon count C to C = C+N−1. Processing then reverts to the first processing step S1 in order to judge whether or not the new as-divided polygons should be sub-divided further. The processing steps S1 to S3 are repeatedly carried out until the new as-divided polygons are comprised within the range of the texture cache.

[0060]

If the result of decision at the first processing step S1 is YES, that is if there is no necessity of dividing the polygon, processing transfers to the next processing step S4.

[0061]

At this processing step S4, the pre-processing information for one polygon is handed over to the polygon engines 33A1, 33A2, ..., 33AN to start rendering processing. Then, processing transfers to the next processing step S5 without waiting for the end of the rendering processing.

[0062]

At this processing step S5, the polygon count C is decremented.

[0063]

At the next processing step S6, it is judged whether or not

the polygon count C is equal to '0'. If the result of processing at this next processing step S6 is NO, with $C \neq 0$, with there being a polygon for processing, processing reverts to the first processing step S1 to enter into the processing of the next polygon. If the result of processing at this next processing step S6 is YES, that is if all polygons have been rendered such that there is no polygon for division, processing comes to a close.

[0064]

That is, the pre-processor 32 judges whether or not the polygon now processed in the drawing engine 33 is comprised within the texture cache 33F (judgment condition 1). The pre-processor 32 performs division processing based on the results of judgment for dividing the polygon corresponding to the drawing command so that the new as-divided polygons will be comprised within the texture cache 33F. This enables texture mapping processing to be performed reliably and efficiently based on the texture data read out in the drawing engine 33 from the texture cache 33F via the CLUT cache 33G.

[0065]

In the division processing of the polygon by the pre-processor 32, it is possible to check in the above-mentioned first processing step whether or not the number of pixels in the polygon is smaller than a prescribed value (judgment condition 2) for checking whether or not the polygon needs to be divided

and to divide the polygon corresponding to the drawing command into plural polygons in a two-dimensional space at the processing step S2 so that the number of pixels in the new as-split polygons will be not larger than the prescribed value. In this manner, the size of the polygon to be processed by the drawing engine can be equated. The number of pixels in the polygon can also be judged by finding the area as an outer product value of the apex points of the polygon and by checking whether or not the value is smaller than an optimum value.

[0066]

In the division processing of the polygon by the pre-processor 32, the polygon corresponding to the drawing command can be divided into plural polygons in a three-dimensional space at the above-mentioned processing step S2.

[0067]

In this case, it is possible to judge at the first processing step S1 whether or not the difference between the minimum value and the maximum value of the Z-value of the apex point of the polygon is comprised within an optimum range (judgment condition 3) in order to check whether or not the polygon needs to be divided and to divide the polygon corresponding to the drawing command at the processing step S2 into plural polygons in the three-dimensional space so that the number of pixels in the new as-divided polygons will be comprised within the prescribed range for limiting the size of the polygon

for executing texture mapping processing in a state with only little texture distortion on the basis of texture data read out from the texture cache 33F via the CLUT cache 33G.

[0068]

In this case, it is possible to judge at the first processing step S1 whether or not the MIP map texture referred to by the minimum and maximum values of the Z-values of the apex points of the polygon is traversed (judgment condition 4) and, based on the results of judgement, to divide the polygon corresponding to the drawing command at the processing step S2 into plural polygons in a three-dimensional space for limiting the range of reference of the MIP mapping referred to in a polygon for efficient MIP mapping based on the MIP map texture data read out from the texture cache 33F via the CLUT cache 33G.

[0069]

It is also possible to judge whether or not the polygon needs to be divided depending on whether or not the number of pixels in a polygon is not more than a pre-set value and to divide the polygon corresponding to the drawing command is not larger than the prescribed value and to divide the polygon corresponding to the drawing command into plural polygons at the processing step S2 in the three-dimensional space.

[0070]

It is similarly possible to predict the drawing processing time for the drawing engine 33 based on, for example, the number

of pixels in the polygon, judge whether or not the polygon needs to be divided based on whether the pre-processing time by the pre-processor 32 is balanced to the drawing processing time by the drawing engine 33, at the processing step S1 (judgment condition 5) and to divide the polygon corresponding to the drawing command at the processing step S2 based on the result of judgment so that the pre-processing time by the pre-processor 32 is balanced to the drawing processing time by the drawing engine 33. This renders it possible to balance the processing time by the pre-processor 32 and that by the drawing engine 33 relative to each other and to construct the pre-processor 32 and the drawing engine 33 by pipelining to realize high-speed drawing efficiently.

[0071]

It is also possible to judge at the processing step S1 whether or not the shape of the polygon processed by the processing engine 33 is suited pixel interleaving (judgment condition 6) at the above-mentioned processing step S1 and to divide the polygon corresponding to the drawing command by the processing step S2 into plural new polygons having the shape suitable for the pixel interleaving. This renders it possible to access the frame buffer 18 efficiently by the drawing engine 33 to effect high-speed drawing processing.

[0072]

It is likewise possible to judge at the processing step S1

whether or not, based on the combination of the above-mentioned various judgment conditions, the polygon needs to be divided, and to divide the polygon corresponding to the drawing command by the processing step S2 into plural new polygons so that the new as-divided polygons meet the judgment conditions.

[0073]

Specifically, it is possible to judge at the processing step S1 whether or not, based on the combination of the judgment conditions 1 and 2, the polygon needs to be divided, and to divide the polygon corresponding to the drawing command by the processing step S2 into plural new polygons so that the new as-divided polygons meet the judgment conditions 1 and 2 in order to equate the size of the polygons processed by the drawing engine 33 and in order to carry out the texture mapping processing reliably and efficiently based on the texture data read out from the texture cache 33F via CLUT cache 33G.

[0074]

It is also possible to judge at the processing step S1 whether or not, based on the combination of the judgment conditions 1 and 3, the polygon needs to be divided, and to divide the polygon corresponding to the drawing command by the processing step S2 into plural new polygons so that the new as-divided polygons meet the judgment conditions 1 and 3. This renders it possible to perform texture mapping processing reliably and efficiently with only little texture distortion

based on texture data read out from the texture cache 33F via CLUT cache 33G. Also, if the judgment condition 2 is combined with the above combination, the size of the polygon processed by the drawing engine 33, that is the number of pixels, can be equated in order to carry out texture mapping.

[0075]

It is also possible to judge at the processing step S1 whether or not, based on the combination of the judgment conditions 1 and 4, the polygon needs to be divided, and to divide the polygon corresponding to the drawing command by the processing step S2 into plural new polygons so that the new as-divided polygons meet the judgment conditions 1 and 4. This renders it possible to perform MIP mapping processing reliably and efficiently based on texture data read out from the texture cache 33F via CLUT cache 33G. Also, if the judgment conditions 2 and 3 are combined with the above combination, it becomes possible to equate the size of the polygon processed by the drawing engine 33, that is the number of pixels, and to diminish texture distortion.

[0076]

It is also possible to judge at the processing step S1 whether or not, based on the combination of the judgment conditions 1 and 5, the polygon needs to be divided, and to divide the polygon corresponding to the drawing command by the processing step S2 into plural new polygons so that the new as-

divided polygons meet the judgment conditions 1 and 5. This renders it possible to keep the processing time by the pre-processor 32 balanced with that by the drawing engine 33 to effect efficient high-speed texture mapping by pipelining. Also, if the judgment conditions 2 and 3 are combined with the above combination, it becomes possible to equate the size of the polygon processed by the drawing engine 33, that is the number of pixels, and to diminish texture distortion. The judgment condition 4 may also be combined with the above combination to effect MIP mapping.

[0077]

It is also possible to judge at the processing step S1 whether or not, based on the combination of the judgment conditions 1 and 6, the polygon needs to be divided, and to divide the polygon corresponding to the drawing command by the processing step S2 into plural new polygons so that the new as-divided polygons meet the judgment conditions 1 and 6. This renders it possible to effect texture mapping reliably and efficiently by the drawing engine 33 and to access the frame buffer 18 efficiently to realize high-speed drawing. Also, if the judgment conditions 2 and 3 are combined with the above combination, it becomes possible to equate the size of the polygon processed by the drawing engine 33, that is the number of pixels, and to diminish texture distortion. The judgment condition 4 may also be combined with the above combination to

effect MIP mapping or the judgment condition 5 may be combined with the above combination to effect high-speed processing by pipelining.

[0078]

The pixel interleaving by the above-mentioned second bus switcher 33E is carried out as described below:

[0079]

Referring to Fig.6, the second bus switcher 33E includes a control circuit 101 fed with an output of the pre-processor 32 shown in Fig.2, a selector 102 fed with an output of the control circuit 101 and plural multiplexer/demultiplexers (MUX/DMUX) 103a, 103b, 103c, 103d, ... each fed with an output of the selector 102.

[0080]

The MUX/DMUX 103a, 103b, 103c, 103d, ... are connected to the frame buffer 18 and to the drawing engine 33 shown in Fig.2.

[0081]

The frame buffer 18 is made up of plural memory banks [1], [2], ..., [X], ..., [L], as shown in Fig.2. Each of the memory banks [1], [2], ..., [X], ..., [L] is made up of a rectangular area represented by 16 addresses (interleaving patterns), so that the 16 addresses can be accessed simultaneously.

[0082]

Therefore, the memory bank [X], for example, of the frame buffer 18, has 16 input/output ports P_0 to P_{15} for accessing the

addresses A_0 to A_{15}. Of the MUX/DMUX 103a, 103b, 103c, 103d, ... , the four MUX/DMUX 103a, 103b, 103c, 103d are each connected to 16 input/output ports P_0 to P_{15}.

[0083]

Also, the four MUX/DMUX 103a, 103b, 103c, 103d are connected in association with four pixel engines $33D_{x1}$, $33D_{x2}$, $33D_{x3}$ and $33D_{x4}$ of the drawing engine 33.

[0084]

Since the memory banks other than the memory bank [X] are configured similarly to the above-mentioned memory bank [X], the detailed description therefor is not made for simplicity. The accessing operation performed by the second bus switcher 33E on the other memory banks is similar to that performed by the second bus switcher 33E on the memory bank [X] as later explained. Therefore, in the following description, only the accessing operation performed by the second bus switcher 33E on the memory bank [X] is explained.

[0085]

First, a series of operations by the second bus switcher 33E is explained.

[0086]

If, for example, the shape of a polygon drawn on the memory bank [X] is a triangle T_{ABC} (the shape of the first polygon) as shown in Fig.7, the control circuit 101 is first fed from the pre-processor 32 with the pixel interleaving control information.

[0087]

Based on the pixel interleaving control information from the pre-processor 32, the control circuit 101 changes over an interleaving pattern, used for accessing the inside of the triangle T_{ABC}, to, for example, a (4×4) interleaving pattern.

[0088]

The method for changing over the interleaving pattern in the control circuit 101 will be later explained in detail.

[0089]

Of plural interleaving patterns formed on the memory bank [X], such interleaving pattern to be accessed, that is, such interleaving pattern as permits the inside of the triangle T_{ABC} to be accessed in its entirety, is detected by the control circuit 101, with the aid of the (4×4) interleaving pattern.

[0090]

Thus, in the triangle T_{ABC}, if each interleaving pattern on the memory bank [X] is indicated by P (pattern index in the x-direction and pattern index in the y-direction), a sum total of 20 interleaving patterns specified by

$$
\begin{aligned}
P(x,y) = \; & P(3,1),\ P(4,1),\ P(1,2),\ P(2,2), \\
& P(3,2),\ P(4,2),\ P(1,3),\ P(2,3), \\
& P(3,3),\ P(4,3),\ P(5,3),\ P(2,4), \\
& P(3,4),\ P(4,4),\ P(5,4),\ P(3,5), \\
& P(4,5),\ P(5,5),\ P(4,6),\ P(5,6)
\end{aligned}
$$

are detected, as shown in Fig.8.

[0091]

The control circuit 101 routes the pattern information specifying the 20 interleaving patterns, detected as described above, to the selector 102 on the interleaving pattern basis. When performing memory accessing on the address basis, the control circuit 101 routes the mask information corresponding to the shape of the triangle T_{ABC} to the selector 102.

[0092]

Based on the pattern information, supplied from the control circuit 101 on the interleaving patten basis, the selector 102 designates the address corresponding to the (4×4) interleaving pattern P to be accessed for the MUX/DMUX 103a, 103b, 103c and 103d.

[0093]

If fed with the mask information from the control circuit 101, the selector 102 designates for MUX/DMUX 103a to 103d the accessing addresses obtained as a result of masking performed in the (4×4) interleaving patterns P, on the basis of the masking information, as shown in Fig.9. Thus, of the addresses A_0 to A_{15} in the interleaving pattern specified by P(4,1) shown in Fig.9, the addresses to be accessed, obtained as the result of masking, are A4, A5, A6, A8, A9, A10, A13, A14 and A15, shown shaded in Fig.10.

[0094]

The MUX/DMUX 103a, 103b, 103c and 103d access the addressees

A_0 to A_{15} on the memory bank [X] designated by the selector 102.

[0095]

The pixel engines $33D_{x1}$, $33D_{x2}$, $33D_{x3}$ and $33D_{x4}$ output pixel data to the MUX/DMUX 103a, 103b, 103c and 103d, respectively, as explained previously.

[0096]

Thus the MUX/DMUX 103a accesses the address designated by the selector 102 to write pixel data from the pixel engine Xa, via one of the input/output ports P_0 to P_{15} corresponding to the address designated by the selector 102, in an area on the memory bank [X] designated by the above address.

[0097]

The MUX/DMUX 103a accesses the address designated by the selector 102 to read out data written in an area designated by the address on the memory bank [X] via one of the input/output ports P_0 to P_{15} corresponding to the above address. The MUX/DMUX 103a performs pre-set processing on the data read out from the memory bank [X].

[0098]

Since the operation of the MUX/DMUX 103b to 103d is similar to the above-described operation of the MUX/DMUX 103a, detailed description therefor is omitted for clarity.

[0099]

The method for changing over the interleaving patterns in the above-described control circuit 101 is now specifically

explained.

[0100]

First, the number of times of accessing the inside of a transversely elongated triangle T_{DEF} (shape of a second polygon) as the shape of a polygon drawn on the memory bank [X] shown in Fig.11 with a (4×4) interleaving pattern P is explained.

[0101]

In this case, the interleaving patterns to be accessed are:

P(x,y) = P(1,1), P(2,1), P(3,1),
P(4,1), P(5,1), P(0,2),
P(1,2), P(2,2), P(3,2),
P(4,2), P(5,2), P(6,2),
P(7,2), P(8,2), P(7,3),
P(8,3), P(9,3),

totaling at 17, as shown in Fig.12.

[0102]

That is, for accessing the inside of the triangle T_{DEF} with the (4×4) interleaving pattern, the number of times of accessing for accessing the inside of the triangle T_{DEF} in its entirety is 17.

[0103]

In case of accessing on the address basis, only the required memory addresses can be accessed by carrying out the masking in the (4×4) interleaving pattern P, as in case of accessing the above-mentioned triangle T_{ABC}, as shown in Fig.13.

[0104]

Then, if the inside of the triangle T_{DEF} is accessed with a (8×2) interleaving pattern P_1 as shown in Fig.14, the interleaving patterns to be accessed are

$$
\begin{aligned}
P_1(x,y) = {} & P_1(1,2),\ P_1(2,2),\ P_1(0,3),\\
& P_1(1,3),\ P_1(2,3),\ P_1(0,4),\\
& P_1(1,4),\ P_1(2,4),\ P_1(3,4),\\
& P_1(1,5),\ P_1(2,5),\ P_1(3,5),\\
& P_1(4,5),\ P_1(3,6),\ P_1(4,6)
\end{aligned}
$$

totaling at 15, as shown in Fig.15.

[0105]

That is, if the inside of the triangle T_{DEF} is accessed with the (8×2) interleaving pattern, the number of times of accessing required for accessing the entire inside of the triangle T_{DEF} is 15.

[0106]

In case of accessing on the address basis, masking is performed within the (8×2) interleaving pattern P_1 as in case of accessing the triangle T_{ABC} as described above, as shown in Fig.16, for accessing only the needed memory address.

[0107]

Then, if the inside of the triangle T_{DEF} is accessed with a (16×1) interleaving pattern P_2, as shown in Fig.17, the interleaving patterns to be accessed are

$P_2(x,y) = P_2(0,5),\ P_2(1,5),\ P_2(0,6),$

$P_2(1,6)$, $P_2(0,7)$, $P_2(1,7)$,

$P_2(0,8)$, $P_2(1,8)$, $P_2(0,9)$,

$P_2(1,9)$, $P_2(0,10)$, $P_2(1,10)$,

$P_2(2,10)$, $P_2(1,11)$, $P_2(2,11)$,

$P_2(1,12)$, $P_2(2,12)$, $P_2(2,13)$

totaling at 18, as shown in Fig.18.

[0108]

That is, if the inside of the triangle T_{DEF} is accessed with the (16×1) interleaving pattern, the number of times of accessing required for accessing the entire inside of the triangle T_{DEF} is 18.

[0109]

In case of accessing on the address basis, masking is performed within the (8×2) interleaving pattern P_2 as in case of accessing the triangle T_{ABC} as described above, as shown in Fig.19, for accessing only the needed memory address.

[0110]

As described above, the number of times of accessing the inside of the triangle T_{DEF} with the (4×4) interleaving pattern P is 17, while that of accessing the inside of the triangle T_{DEF} with the (8×2) interleaving pattern P_1 is 15 and that of accessing the inside of the triangle T_{DEF} with the (16×1) interleaving pattern P_2 is 18. Thus, the number of times of accessing the inside of the triangle T_{DEF} with the (8×2)

interleaving pattern P_1 represents the minimum number of times of accessing. Thus it is seen that a proper interleaving pattern for the triangle T_{DEF} is the (8×2) interleaving pattern P_1.

[0111]

Thus, for switching the interleaving pattern used for accessing the memory bank [X] to a proper interleaving pattern in meeting with the shape of the polygon to be accessed, the control circuit 101 executes the following processing operations.

[0112]

If, for example, the shape of a polygon drawn on the memory bank [X] is a triangle T_{HIJ}, as shown in Fig.20, the control information for pixel interleaving is supplied from the pre-processor 32 to the control circuit 101, as described above. For example, the control information for pixel interleaving is such information as xy coordinates H (Xh, Yh), I (Xi, Yi) or J (Xj, Yj) of three apex points of the triangle T_{HIJ}.

[0113]

The control circuit 101 then finds, using the control information for pixel interleaving from the pre-processor 32, the aspect ratio R of the triangle T_{HIJ}, by calculations:

$$R = dy/dx$$
$$= (MAXy - MINy)/(MAXx - MINx)$$

where MAXx and MINx are maximum and minimum values in the X-direction, respectively, and MAXy and MINy are maximum and minimum values in the Y-direction, respectively, as shown in

Fig.20.

[0114]

In the triangle T_{HIJ},

$MAXx = Xj$

$MINx = Xi$

$MAXy = Yh$

$MINy = Yi$

[0115]

Depending on the aspect ratio R, thus found, the control circuit 101 selects one of five interleaving patterns Pa to Pe of (1×16), (2×8), (4×4), (8×2) and (16×1), as shown in Fig.21, and switches the interleaving pattern used in accessing the inside of the triangle T_{HIJ} to the selected interleaving pattern.

[0116]

The control circuit 101 has a table for the aspect ratio R and interleaving patterns (Table 1). In this table are pre-set proper interleaving patterns associated with various values of the aspect ratio R, that is such an interleaving pattern as minimizes the number of times of accessing. Thus the control circuit 101 selects, using the above table, a proper interleaving pattern associated with the aspect ratio R found as described above.

[0117]

[TABLE 1]

aspect ratio R	interleaving pattern
~ 0.1	Pa (16×1)
0.1 ~ 0.5	Pb (8×2)
0.5 ~ 2.0	Pc (4×4)
2.0 ~ 8.0	Pd (2×8)
8.0 ~	Pe (1×16)

[0118]

In the second bus switcher 33E, a proper one of five interleaving patterns Pa to Pe shown in Fig.21 is selected responsive to the shape of the polygon to be drawn on the memory bank [X], which is then accessed with the selected interleaving pattern, so that the polygon can be drawn on the memory bank [X] with the minimum number of accessing operations. Thus it is possible for the second bus switcher 33E to effect memory accessing efficiently.

[0119]

The GPU 15 accesses the frame buffer 18 by the second bus switcher 33E aimed at raising the memory accessing efficiency, for performing various data processing operations, as described above, for realizing efficient data processing operations.

[0120]

[Effect of the Invention]

With the picture drawing device and method according to the present invention, texture data required for texture mapping by the picture drawing means is transmitted from a texture memory to a texture cache in a pre-processing stage by the pre-processing means. The pre-processing means and picture drawing means are operated in pipelining for enabling picture drawing without halting the picture drawing means.

[0121]

With the picture drawing device and method according to the

present invention, data of the resolution required for the picture drawing means to perform the MIP mapping is selected from the texture data on the texture memory in the pre-processing stage by the pre-processing means and transmitted to the texture cache for reducing the number of times of accessing and the accessing time for the texture memory for raising the overall picture drawing speed.

[Brief Description of the Drawings]

Fig.1 is a block diagram showing the structure of a video game machine embodying the present invention.

Fig.2 is a block diagram showing a detailed structure of a GPU in the video game machine.

Fig.3 is a block diagram showing the basic structure of the GPU.

Fig.4 shows an example of a data structure in a texture cache in the GPU.

Fig.5 is a flowchart showing the processing of resolving a first polygon by a pre-processor in the GPU.

Fig.6 is a block diagram showing the structure of a second bus switcher in the video game machine.

Fig.7 illustrates the case of accessing the inside of the shape of a first polygon drawn on the memory bank of the frame buffer in the video game machine.

Fig.8 illustrates an interleaving pattern to be accessed when having access to the inside of the shape of the first

polygon.

Fig.9 illustrates masking in case of for address-based accessing when having access to the inside of the shape of the first polygon.

Fig.10 illustrates the accessing address obtained by the masking.

Fig.11 illustrates accessing to the inside of the shape of the second polygon drawn on a memory bank of the frame buffer with a (4×4) interleaving pattern.

Fig.12 illustrates an interleaving pattern to be accessed when accessing to the inside of the shape of the second polygon drawn on a memory bank of the frame buffer with a (4×4) interleaving pattern.

Fig.13 illustrates masking in case of address-based accessing to the inside of the shape of the second polygon with a (4×4) interleaving pattern.

Fig.14 illustrates the inside of the second polygon with a (8×2) interleaving pattern.

Fig.15 illustrates an interleaving pattern to be accessed when accessing to the inside of the shape of the second polygon drawn on a memory bank of the frame buffer with a (8×2) interleaving pattern.

Fig.16 illustrates masking in case of address-based accessing to the inside of the shape of the second polygon with a (8×2) interleaving pattern.

Fig.17 illustrates the inside of the second polygon with a (16×1) interleaving pattern.

Fig.18 illustrates an interleaving pattern to be accessed when accessing to the inside of the shape of the second polygon drawn on a memory bank of the frame buffer with a (16×1) interleaving pattern.

Fig.19 illustrates masking in case of address-based accessing to the inside of the shape of the second polygon with a (16×1) interleaving pattern.

Fig.20 illustrates processing for calculating the aspect ratio of a polygon drawn on a memory bank of the frame buffer.

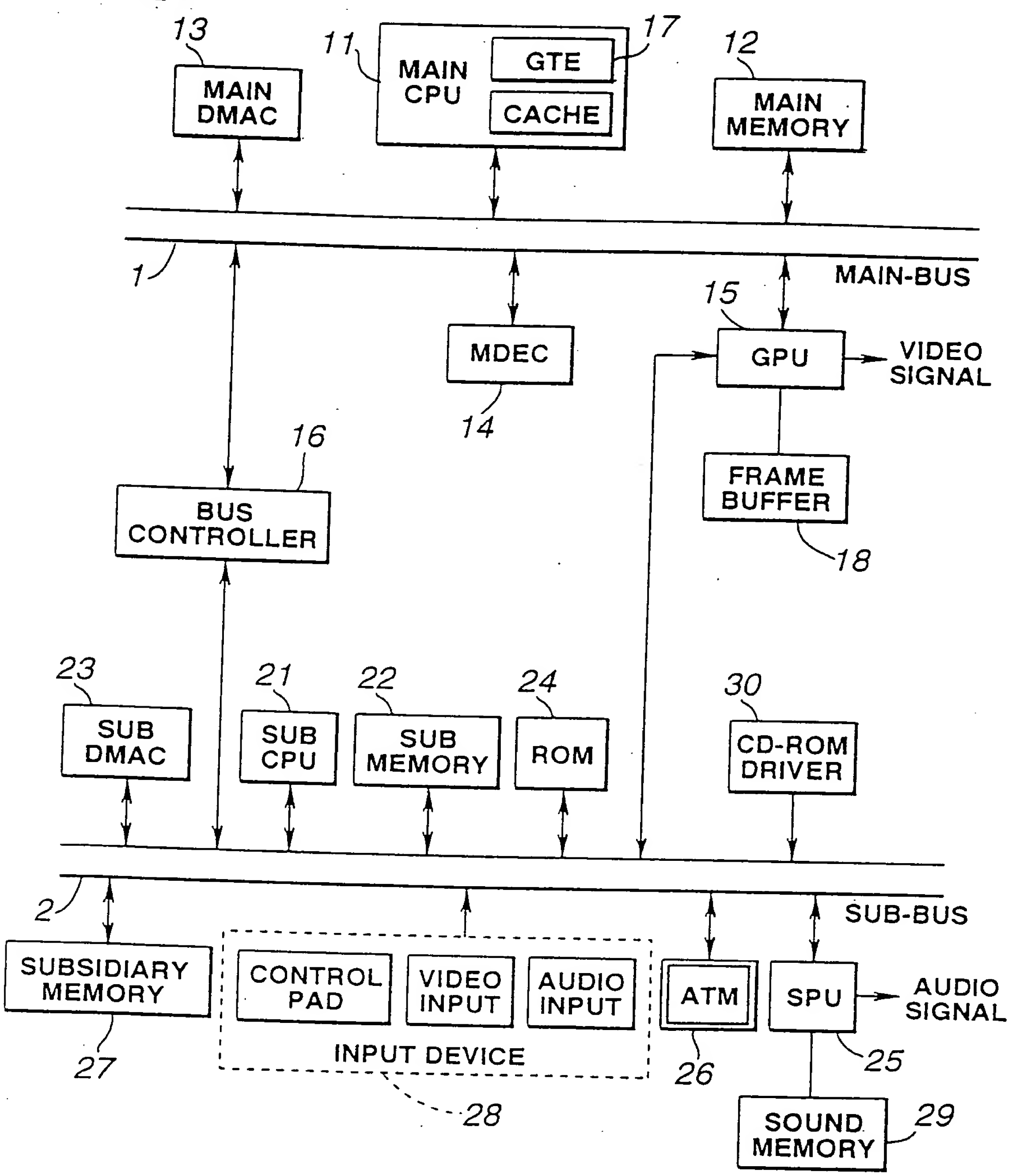
Fig.21 is a pattern diagram showing five sorts of interleaving patterns having 16 addresses.

[Description of the Numerals]

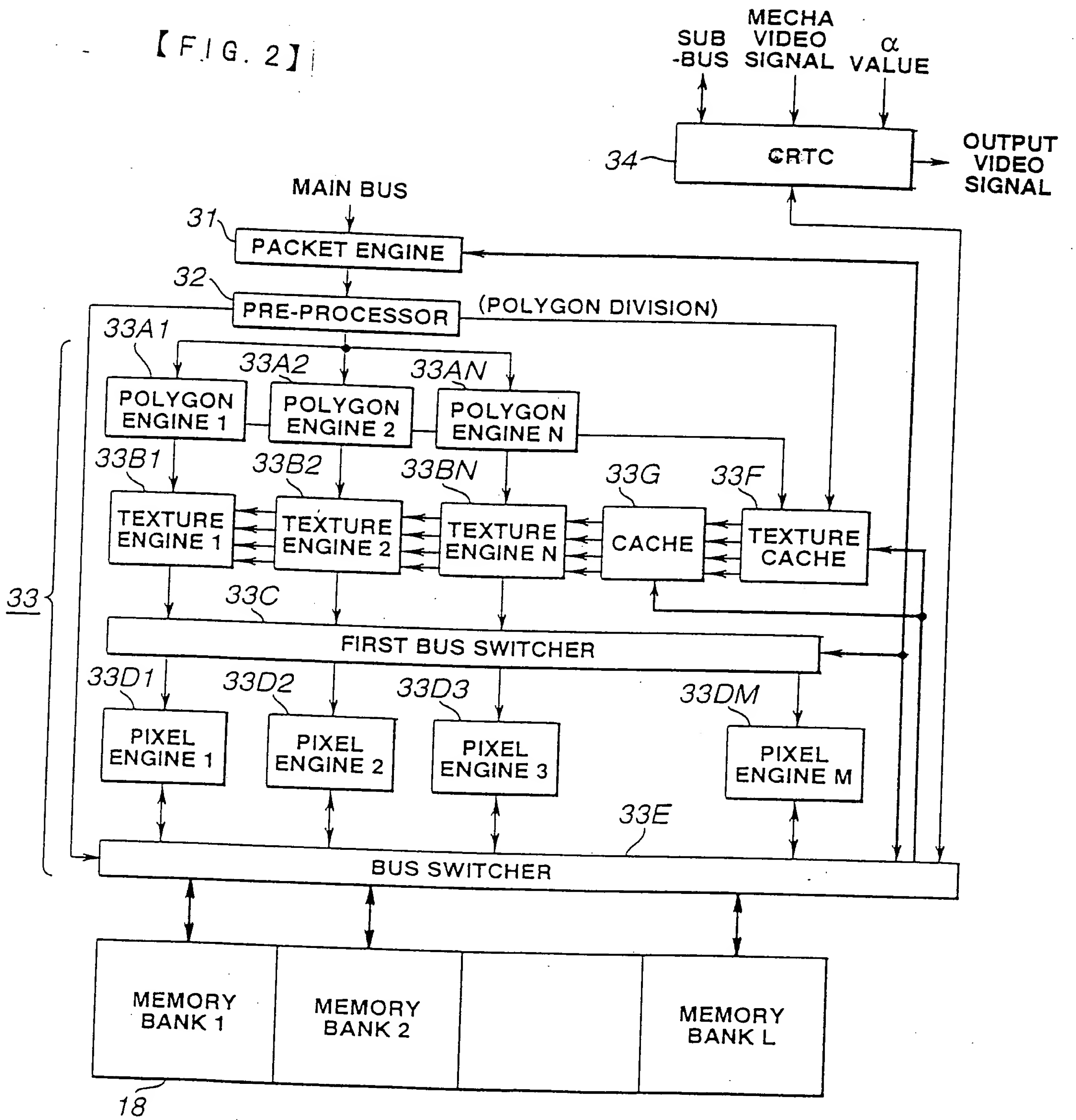
1 main bus; 11 main CPU; 12 main memory; 13 main DMAC; 15 GPU; 17 GTE; 18 frame buffer; 31 packet engine; 32 pre-processor; 33 drawing engine; 33A1, 33A2 ... 33AN polygon engine; 33B1, 33B2 ... 33BN texture engine; 33C first bus switcher; 33D1, 33D2 ... 33DM, 33E second bus switcher; 33F texture cache

【DOCUMENT NAME】 DRAWING

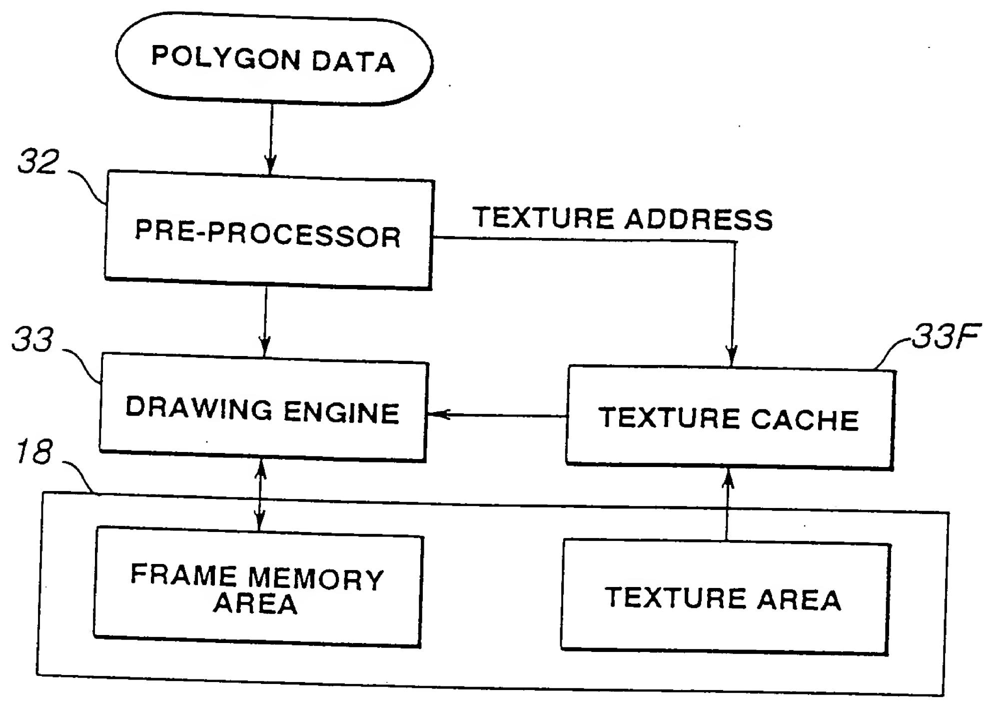
【FIG. 1】



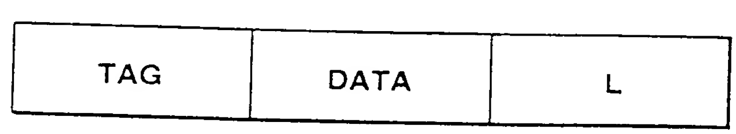
【FIG. 2】


SUB -BUS
MECHA VIDEO SIGNAL
α VALUE
34
CRTC
OUTPUT VIDEO SIGNAL
MAIN BUS
31
PACKET ENGINE
32
PRE-PROCESSOR
(POLYGON DIVISION)
33A1
33A2
33AN
POLYGON ENGINE 1
POLYGON ENGINE 2
POLYGON ENGINE N
33B1
33B2
33BN
33G
33F
TEXTURE ENGINE 1
TEXTURE ENGINE 2
TEXTURE ENGINE N
CACHE
TEXTURE CACHE
33
33C
FIRST BUS SWITCHER
33D1
33D2
33D3
33DM
PIXEL ENGINE 1
PIXEL ENGINE 2
PIXEL ENGINE 3
PIXEL ENGINE M
33E
BUS SWITCHER
MEMORY BANK 1
MEMORY BANK 2
MEMORY BANK L
18

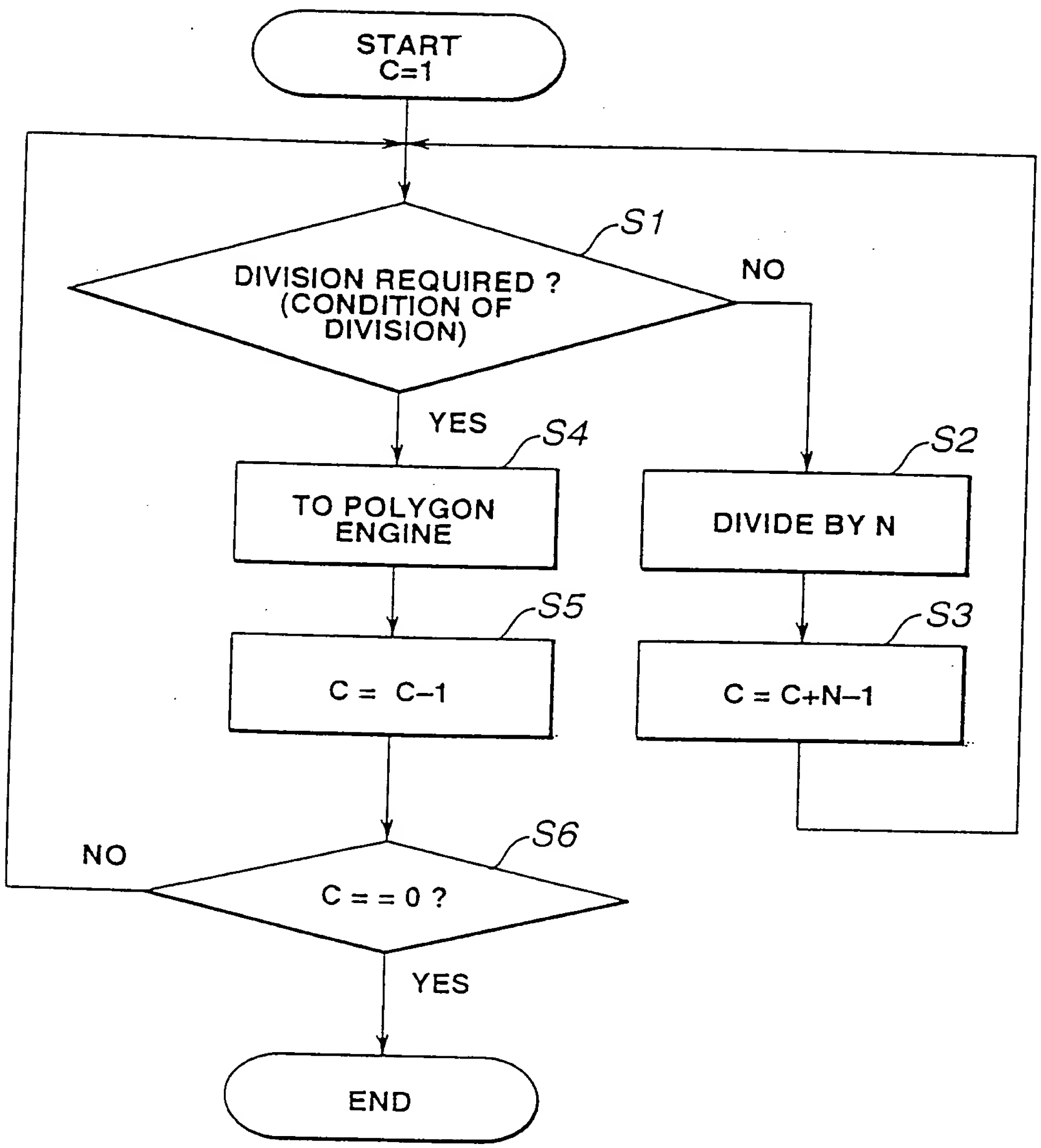
【FIG. 3】


POLYGON DATA
32
PRE-PROCESSOR
TEXTURE ADDRESS
33
DRAWING ENGINE
33F
TEXTURE CACHE
18
FRAME MEMORY AREA
TEXTURE AREA

【FIG. 4】

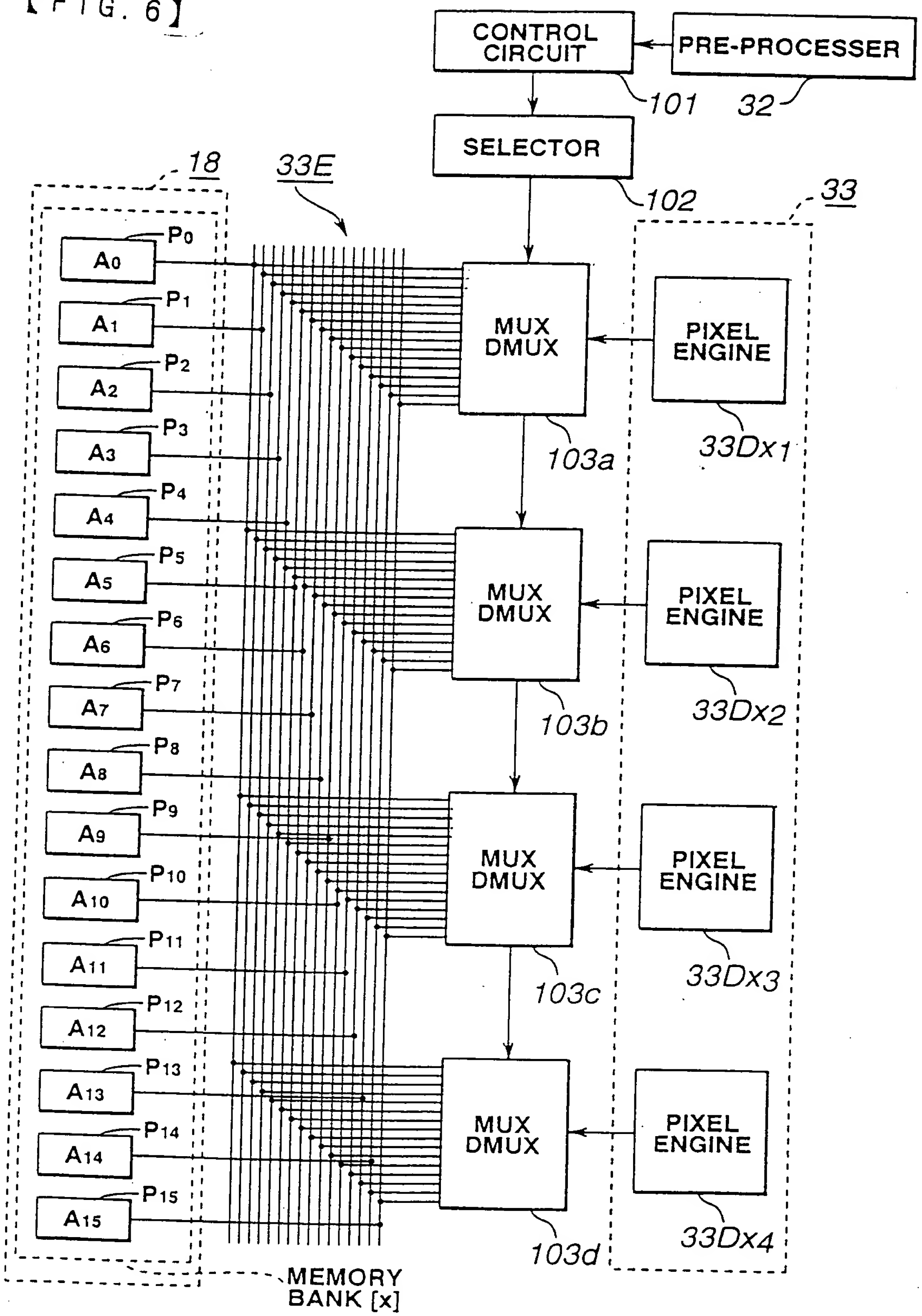

TAG
DATA
L

【FIG. 5】



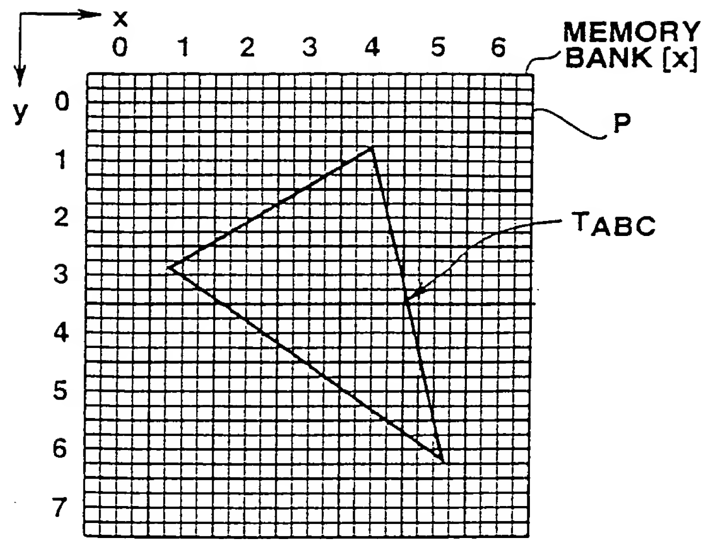
START
C=1
S1
DIVISION REQUIRED ?
(CONDITION OF
DIVISION)
NO
YES
S4
TO POLYGON
ENGINE
S2
DIVIDE BY N
S5
C = C–1
S3
C = C+N–1
S6
NO
C = = 0 ?
YES
END

【FIG. 6】



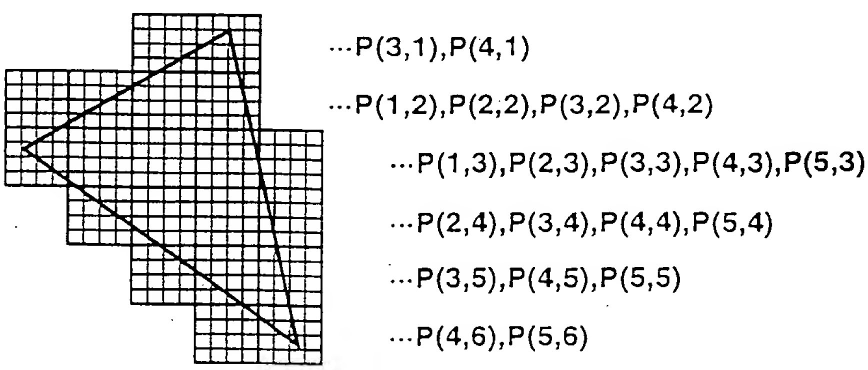
BUS SWITCHER EMPLOYING MEMORY ACCESS METHOD ACCORDING TO THIS INVENTION

【FIG. 7】



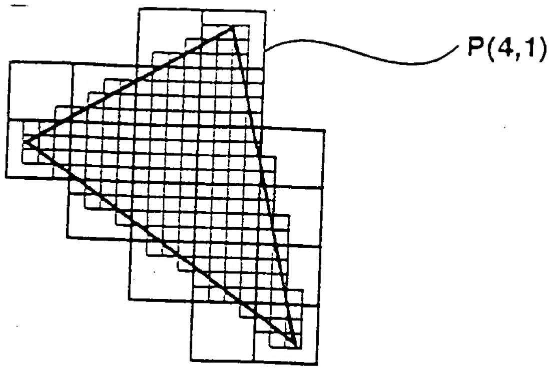
SHAPE OF A FIRST POLYGON DRAWN ON MEMORY BANK

【FIG. 8】



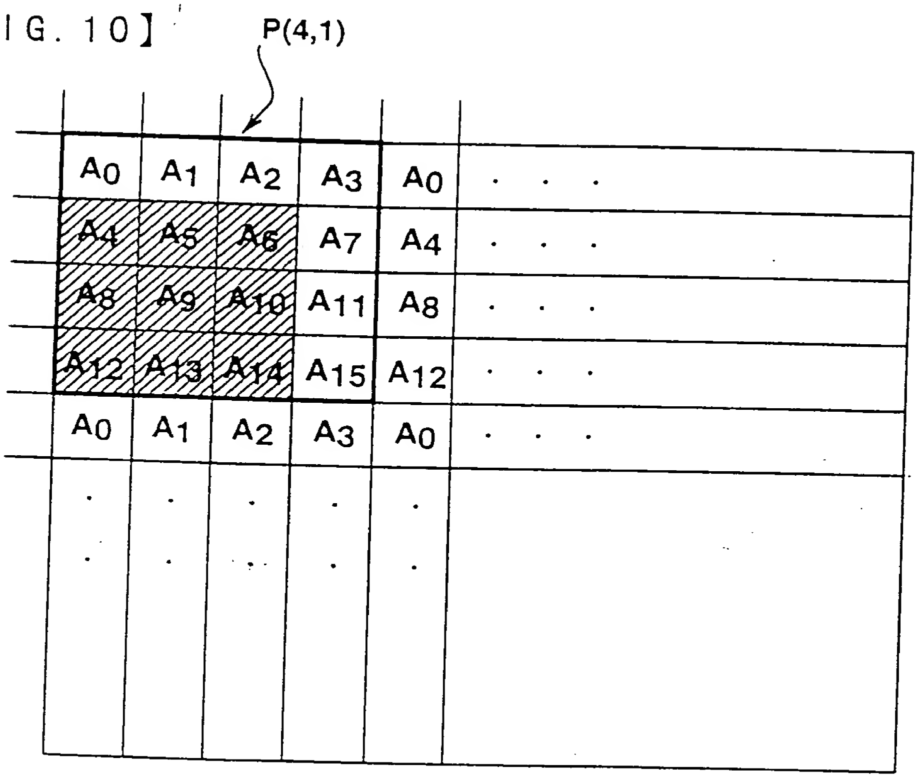
INTERLEAVING PATTERN TO BE ACCESSED

【FIG. 9】



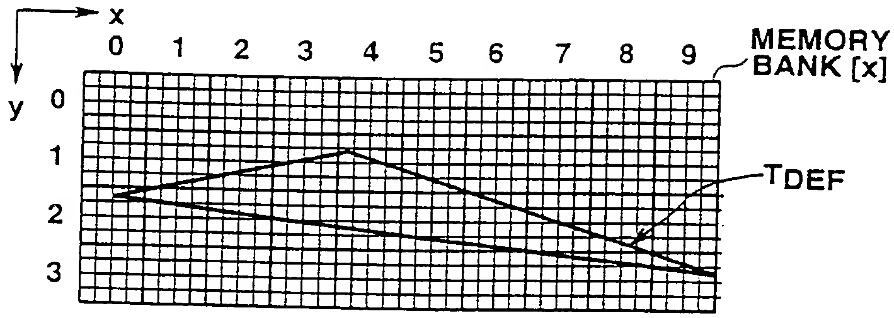
MASKING IN CASE OF FOR ADDRESS-BAND ACCESSING

【FIG. 10】



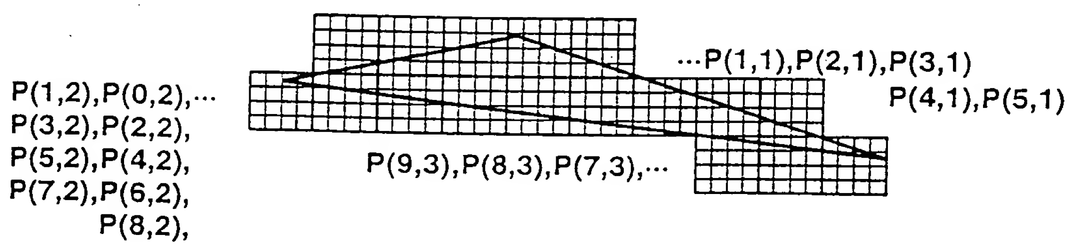
ACCESSING ADDRESS OBTAINED BY MASKING

【FIG. 11】



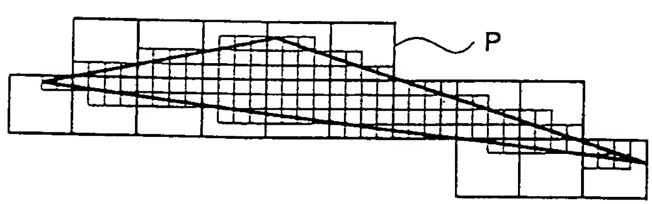
ACCESSING TO SHAPE OF THE SECOND POLYGON WITH (4×4) INTERLEAVING PATTERN

【FIG. 12】



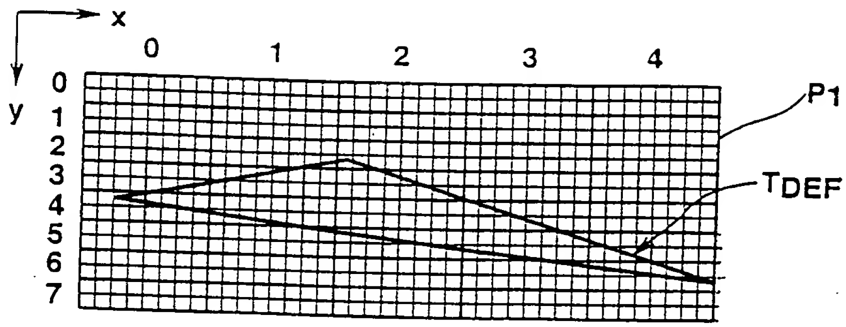
INTERLEAVING PATTERN TO BE ACCESSED WITH (4×4) INTERLEAVING PATTERN

【FIG. 13】



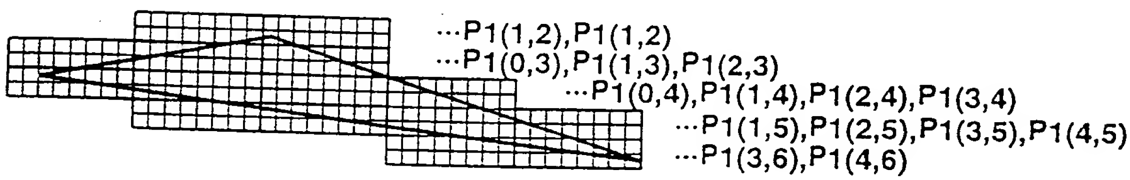
MASKING IN CASE OF ADDRESS-BASED ACCESSING WITH (4×4) INTERLEAVING PATTERN

【FIG. 14】



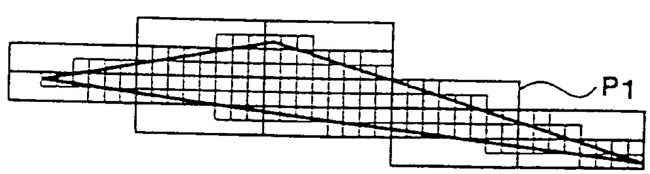
ACCESSING TO SHAPE OF THE SECOND POLYGON
WITH (8×2) INTERLEAVING PATTERN

【FIG. 15】



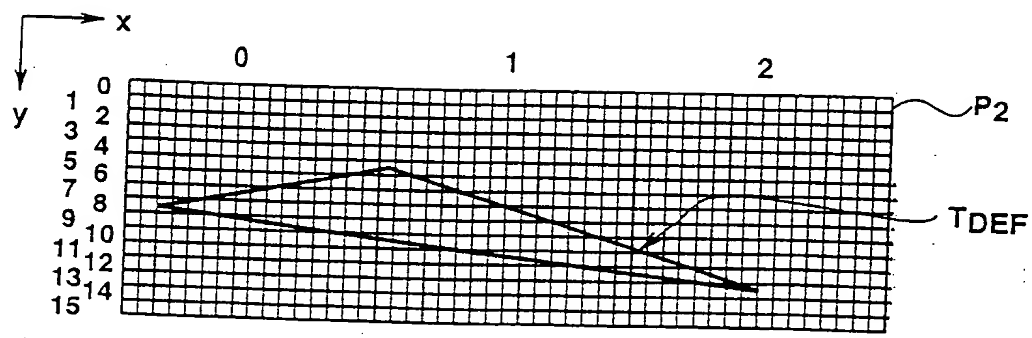
INTERLEAVING PATTERN TO BE ACCESSED
WITH (8×2) INTERLEAVING PATTERN

【FIG. 16】



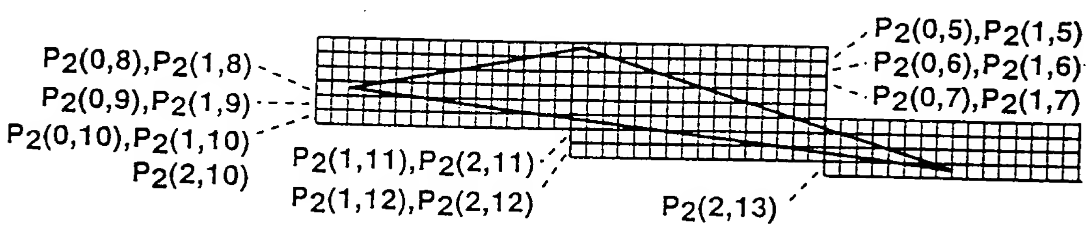
MASKING IN CASE OF ADDRESS-BASED ACCESSING
WITH (8×2) INTERLEAVING PATTERN

【FIG. 17】



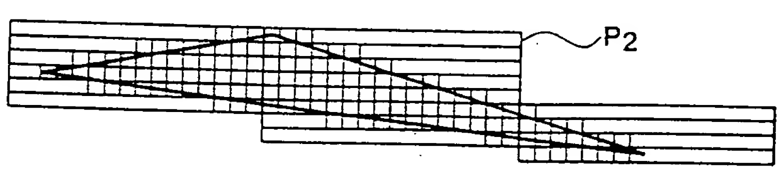
IN CASE OF ACCESSING SHAPE OF THE SECOND POLYGON WITH (16×1) INTERLEAVING PATTERN

【FIG. 18】



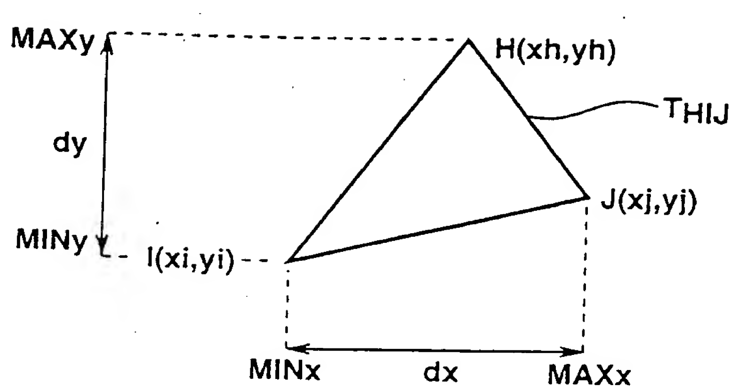
INTERLEAVING PATTERN TO BE ACCESSED WHEN ACCESSED TO WITH (16×1) INTERLEAVING PATTERN

【FIG. 19】



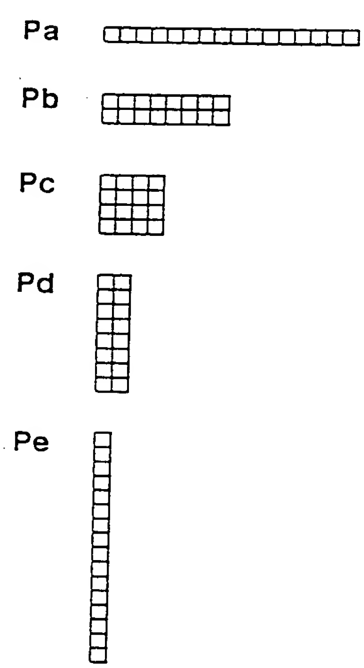
MASKING IN CASE OF ADDRESS-BAND ACCESSING TO WITH a (16×1) INTERLEAVING PATTERN

【FIG. 20】



CALCULATING ASPECT RATION

【FIG. 21】



FIVE SORTS OF INTERLEAVING PATTERNS

[Name of Document] ABSTRACT

[Summary]

[Task]

To provide a picture drawing apparatus and a picture drawing method which enable texture mapping or MIP mapping without halting the picture drawing means, and reduction in the number of times of texture memory accessing and the accessing time for raising the overall picture drawing speed.

[Means for Solution]

A picture drawing or method used in a graphics computer, a special effect device or a video game machine. For generating data required for picture drawing by pre-processing by a pre-processor 32 based on a drawing command for drawing a picture model defined by the combination of unit figures, and generating pixel data on the unit figure basis by texture mapping based on the generated data for drawing a picture on a frame buffer 18, the texture data required by a drawing engine 33 is transferred in the pre-processing stage from a texture area on the frame buffer 18 to a texture cache 33F, and the pre-processor 32 and the drawing engine 33 are operated in pipelining.

[Selected Drawing] Fig.3

[Document Name] Official Correction Data

[Corrected Document] Patent Application

<Authorized Information · Additional Information>

[Patent Applicant]

[Identification Number] 395015319

[Address] 1-22, Akasaka 8-chome, Minato-ku, Tokyo

[Name] Sony Computer Entertainment Inc.

[Agent] Applicant

[Identification Number] 100067736

[Address] A. Koike & Co.
2nd Floor, No.11-Mori Bldg.,
No. 6-4, Toranomon 2-chome,
Minato-ku, Tokyo

[Name] Akira Koike

[Agent]

[Identification Number] 100086335

[Address] A. Koike & Co.
2nd Floor, No.11-Mori Bldg.,
No. 6-4, Toranomon 2-chome,
Minato-ku, Tokyo

[Name] Eiichi Tamura

[Agent]

[Identification Number] 100096677

[Address] A. Koike & Co.
2nd Floor, No.11-Mori Bldg.,
No. 6-4, Toranomon 2-chome,
Minato-ku, Tokyo

[Name] Seiji Iga

Information of Record for Applicant

Identification Number: [395015319]

1. Date of Change: July 21, 1995

[Reason of Change] Registration

[Address] 1-22, Akasaka 8-chome, Minato-ku, Tokyo, Japan

[Name] Sony Computer Entertainment Inc.